Comput Optim Appl (2012) 53:505–526
DOI 10.1007/s10589-011-9405-3

Derivative-free methods for bound constrained mixed-integer optimization

G. Liuzzi · S. Lucidi · F. Rinaldi

Received: 16 November 2010 / Published online: 6 April 2011

Abstract We consider the problem of minimizing a continuously differentiable function of several variables subject to simple bound constraints where some of the variables are restricted to take integer values. We assume that the first order derivatives of the objective function can be neither calculated nor approximated explicitly. This class of mixed integer nonlinear optimization problems arises frequently in many industrial and scientific applications and this motivates the increasing interest in the study of derivative-free methods for their solution. The continuous variables are handled by a linesearch strategy whereas to tackle the discrete ones we employ a local search-type approach. We propose different algorithms which are characterized by the way the current iterate is updated and by the stationarity conditions satisfied by the limit points of the sequences they produce.

Keywords Derivative-free optimization · Bound constrained optimization · Mixed-integer nonlinear programming

G. Liuzzi
Istituto di Analisi dei Sistemi ed Informatica "A. Ruberti", Consiglio Nazionale delle Ricerche, Viale Manzoni 30, 00185 Rome, Italy
e-mail: liuzzi@iasi.cnr.it

S. Lucidi · F. Rinaldi (✉)
Dipartimento di Informatica e Sistemistica "A. Ruberti", "Sapienza" Università di Roma, Via Ariosto 25, 00185 Rome, Italy
e-mail: rinaldi@dis.uniroma1.it

S. Lucidi
e-mail: lucidi@dis.uniroma1.it

1 Introduction

In the paper we consider the following bound constrained mixed variable problem

$$\begin{aligned} &\min f(x) \\ &l \le x \le u, \\ &x_i \in \mathbb{Z} \quad i \in I_z, \end{aligned} \tag{1}$$

where $x \in \mathbb{R}^n$, $l, u \in \mathbb{R}^n$, and $I_z \subseteq \{1, \ldots, n\}$. We assume $l_i < u_i$, for all $i = 1, \ldots, n$, $l_i, u_i \in \mathbb{Z}$, for all $i \in I_z$ and $f : \mathbb{R}^n \to \mathbb{R}$ to be a continuously differentiable function with respect to x_i, $i \notin I_z$. We define the following sets,

$$X = \{x \in \mathbb{R}^n : l \le x \le u\}, \qquad \mathcal{Z} = \{x \in \mathbb{R}^n : x_i \in \mathbb{Z}, i \in I_z\},$$

and assume throughout the paper that X is a compact set, hence l_i and u_i can not be infinite.

In [1, 3, 6, 14] a problem more general than (1) has been considered by allowing also for the presence of categorical variables. The algorithms proposed in the papers [1, 3, 6, 14] are based on the idea of alternating between a local minimization with respect to the continuous variables and a local search with respect to the discrete variables. The common feature of the methods is represented by the fact that the discrete neighborhood structure (that is needed to define the local search) is fixed a priori at every iterate. The cited papers substantially differ in the definition of the continuous minimization phase. In [6] this phase is carried out by a pattern search strategy for box constrained problems [17, 19]. In [14] a linesearch strategy for linearly constrained problems [16] is adopted to carry out the continuous minimization phase. A pattern search strategy combined with a filter approach [5] to tackle general nonlinear constraints has been proposed in [3]. Finally, in [1] for the general nonlinear constrained problem an extreme barrier penalty is adopted and a mesh adaptive direct search strategy [4] is used to force convergence.

In this paper we propose the use of linesearch-type algorithms to solve the problem. For the continuous variables we adopt a well-studied linesearch with sufficient decrease strategy [15]. For the discrete variables we propose the use of different local search procedures. They explore a discrete neighborhood of points whose structure is not defined a priori but it is adaptively determined by a linesearch-type procedure.

The paper is organized as follows. In Sect. 2 we introduce some definitions and relevant notations. Section 3 is the main part of the paper and is devoted to the definition and analysis of three different algorithms for the solution of Problem (1). Finally, in Sect. 5 we draw some conclusions and discuss future developments.

2 Definitions and notations

We introduce the set of indices of continuous variables

$$I_c = \{1, \ldots, n\} \setminus I_z.$$

Given a vector $v \in \mathbb{R}^n$ we introduce the following subvectors $v_c \in \mathbb{R}^{|I_c|}$ and $v_z \in \mathbb{R}^{|I_z|}$ given by

$$v_c = [v_i]_{i \in I_c}, \qquad v_z = [v_i]_{i \in I_z}.$$

For every continuously differentiable function $h : \mathbb{R}^n \to \mathbb{R}$ with respect to the continuous variables, we use the notation $\nabla_c h(x) \in \mathbb{R}^{|I_c|}$ to denote the gradient of the function with respect to the continuous variables, namely:

$$\nabla_c h(x) = \left[\frac{\partial h(x)}{\partial x_i} \right]_{i \in I_c}.$$

We introduce the following definition of neighborhoods with respect to continuous and discrete variables. Given a point $\bar{x} \in \mathbb{R}^n$, let us define

$$\begin{aligned} \mathcal{B}_c(\bar{x}, \rho) &= \{x \in \mathbb{R}^n : x_z = \bar{x}_z, \ \|x_c - \bar{x}_c\|_2 \le \rho\}, \\ \mathcal{N}_z(\bar{x}) &= \{x \in \mathbb{R}^n : x_c = \bar{x}_c, \ \|x_z - \bar{x}_z\|_2 = 1\}. \end{aligned}$$

Due to the mixed-integer nature of Problem (1), different definitions of a local minimum point can be envisaged. In this paper, we consider the following definition of minimum points for Problem (1).

Definition 1 (Local minimum point) A point $x^* \in X$ is a local minimum of Problem (1) if, for some $\epsilon > 0$,

$$\begin{aligned} f(x^*) &\le f(x), \quad \forall x \in \mathcal{B}_c(x^*; \epsilon) \cap X, \\ f(x^*) &\le f(x), \quad \forall x \in \mathcal{N}_z(x^*) \cap X, \end{aligned} \tag{2}$$

and, every point $\bar{x} \in \mathcal{N}_z(x^*) \cap X$ such that $f(\bar{x}) = f(x^*)$ satisfies (2) for some $\bar{\epsilon} > 0$.

Proposition 2 *Let $x^* \in X \cap \mathcal{Z}$ be a local minimum of Problem (1). Then*

$$\nabla_c f(x^*)^T (x - x^*)_c \ge 0, \quad \textit{for all } x \in X, \tag{3}$$

$$f(x^*) \le f(x) \quad \textit{for all } x \in \mathcal{N}_z(x^*) \cap X. \tag{4}$$

Proposition 2 essentially states that a minimum point of Problem (1) has to be stationary with respect to the continuous variables and, with respect to the discrete variables, it must be a local minimum within the discrete neighborhood $\mathcal{N}_z(x^*)$.

With reference to Problem (1), we introduce the following definitions of stationary point and strong stationary point.

Definition 3 (Stationary point) A point $x^* \in X \cap \mathcal{Z}$ is a stationary point of Problem (1) when it satisfies (3) and (4).

Definition 4 (Strong stationary point) A point $x^* \in X \cap \mathcal{Z}$ is a strong stationary point of Problem (1) when it satisfies (3) and (4), and, for all $\bar{x} \in \mathcal{N}_z(x^*) \cap X$ such

that $f(\bar{x}) = f(x^*)$, it holds that

$$\nabla_c f(\bar{x})^T (x - \bar{x})_c \geq 0, \quad \text{for all } x \in X, \tag{5}$$

$$f(\bar{x}) \leq f(x), \quad \text{for all } x \in \mathcal{N}_z(\bar{x}) \cap X. \tag{6}$$

We denote

$$D = \{\pm e_1, \ldots, \pm e_n\}, \qquad D^c = \{\pm e_i : \quad i \in I_c\}, \qquad D^z = \{\pm e_i : \quad i \in I_z\}$$

where e_i, $i = 1, \ldots, n$, is the unit coordinate vector.

Given $x \in X$, we denote by

$$L(x) = \{i \in \{1, \ldots, n\} : x_i = l_i\}, \qquad U(x) = \{i \in \{1, \ldots, n\} : x_i = u_i\}.$$

Given $x \in X$, let

$$D(x) = \{d \in \mathbb{R}^n : d_i \geq 0 \ \forall i \in L(x), d_i \leq 0 \ \forall i \in U(x)\}.$$

We report two technical proposition whose proofs can be found, respectively, in [13] and [12].

Proposition 5 *For every $x \in X$, it results*

$$cone\{D \cap D(x)\} = D(x). \tag{7}$$

Proposition 6 *Let $\{x_k\}$ be a sequence of points such that $x_k \in X$ for all k, and $x_k \to \bar{x}$ for $k \to \infty$. Then, for k sufficiently large,*

$$D(\bar{x}) \subseteq D(x_k).$$

3 Algorithms for bound constrained mixed integer nonlinear optimization

This section is devoted to the definition of different algorithms for the solution of Problem (1) and to the analysis of their convergence properties. The first two algorithms are convergent towards stationary points of the problem. The first algorithm, that is called DFL (Derivative-Free Linesearch), explores the coordinate directions and updates the iterate whenever a sufficient reduction of the objective function is found. Hence it performs a minimization distributed along all the variables. The second algorithm, which is called DFL_{ord}, carries out a minimization which is only distributed along the continuous variables while, along the discrete ones, it updates the iterate by choosing the coordinate that yields the largest objective function reduction. The last algorithm, SDFL, is convergent to strong stationary points. To achieve such a result, Algorithm SDFL performs a deeper investigation of the discrete neighborhoods by means of a local search procedure.

3.1 A distributed algorithmic scheme

In this subsection, we define the distributed derivative-free algorithm for bound constrained mixed integer problems. The basic ingredients of the method are the *Continuous search* and *Discrete search* procedures. They are needed to explore the coordinate directions associated with, respectively, continuous and discrete variables. The current point is updated as soon as a sufficient reduction of the objective function is achieved by one of the procedures. The Continuous search procedure is quite standard in a derivative-free context and we refer the interested reader to [15]. The Discrete search procedure is similar to the Continuous search but the sufficient reduction is governed by a control parameter ξ, which is reduced during the optimization process. In particular, the control parameter ξ is reduced whenever no discrete variable has been updated by the Discrete search procedure and the tentative steps along the discrete variables are equal to one. For this reason, the convergence properties of Algorithm DFL can be characterized only with respect to a particular subsequence of iterates. We denote with $\tilde{\alpha}^i$ the tentative steps used to sample the objective function along the direction d^i, with $i = 1, \ldots, n$. Whenever the step must be reduced, a constant factor $\theta \in (0, 1)$ is adopted. Finally, x^0 denotes the starting point. The algorithm is described as follows.

Then we formally define the Continuous search and Discrete search procedures. The Continuous search procedure is defined by specifying values for parameters γ

Algorithm DFL

Data. $\theta \in (0, 1)$, $\xi_0 > 0$, $x_0 \in X \cap \mathcal{Z}$, $\tilde{\alpha}_0^i > 0$, $i \in I_c$, $\tilde{\alpha}_0^i = 1$, $i \in I_z$, and set $d_0^i = e^i$, for $i = 1, \ldots, n$.

For $k = 0, 1, \ldots$
  Set $y_k^1 = x_k$.
  For $i = 1, \ldots, n$
    If $i \in I_c$ **then** compute α by the *Continuous search*$(\tilde{\alpha}_k^i, y_k^i, d_k^i; \alpha)$

      If $\alpha = 0$ **then** set $\alpha_k^i = 0$ and $\tilde{\alpha}_{k+1}^i = \theta \tilde{\alpha}_k^i$.
      else set $\alpha_k^i = \alpha$, $\tilde{\alpha}_{k+1}^i = \alpha$.

    else compute α by the *Discrete Search*$(\tilde{\alpha}_k^i, y_k^i, d_k^i, \xi_k; \alpha)$

      If $\alpha = 0$ **then** set $\alpha_k^i = 0$ and $\tilde{\alpha}_{k+1}^i = \max\{1, \lfloor \tilde{\alpha}_k^i / 2 \rfloor\}$.
      else set $\alpha_k^i = \alpha$, $\tilde{\alpha}_{k+1}^i = \alpha$.

    Set $y_k^{i+1} = y_k^i + \alpha_k^i d_k^i$ and $d_{k+1}^i = d_k^i$.

  End For
  If $(y_k^{n+1})_z = (x_k)_z$ and $\tilde{\alpha}_k^i = 1$, $i \in I_z$, **then** set $\xi_{k+1} = \theta \xi_k$ **else** set $\xi_{k+1} = \xi_k$.
  Find $x_{k+1} \in X \cap \mathcal{Z}$ such that $f(x_{k+1}) \leq f(y_k^{n+1})$.
End For

Continuous search $(\tilde{\alpha}, y, d; \alpha)$

Data. $\gamma > 0$, $\delta \in (0, 1)$.
Step 1. Compute the largest $\bar{\alpha}$ such that $y + \bar{\alpha} d \in X \cap \mathcal{Z}$. Set $\alpha = \min\{\bar{\alpha}, \tilde{\alpha}\}$.
Step 2. If $\alpha > 0$ and $f(y + \alpha d) \leq f(y) - \gamma \alpha^2$ **then** go to Step 6.
Step 3. Compute the largest $\bar{\alpha}$ such that $y - \bar{\alpha} d \in X \cap \mathcal{Z}$. Set $\alpha = \min\{\bar{\alpha}, \tilde{\alpha}\}$.
Step 4. If $\alpha > 0$ and $f(y - \alpha d) \leq f(y) - \gamma \alpha^2$ **then** set $d \leftarrow -d$ and go to Step 6.
Step 5. Set $\alpha = 0$ and **return**.
Step 6. While $\left(\alpha < \bar{\alpha} \text{ and } f\left(y + \frac{\alpha}{\delta} d \right) \leq f(y) - \gamma \frac{\alpha^2}{\delta^2} \right)$
$\alpha \leftarrow \alpha / \delta$.
Step 7. Set $\alpha \leftarrow \min\{\bar{\alpha}, \alpha\}$ and **return**.

Discrete search $(\tilde{\alpha}, y, d, \xi; \alpha)$

Step 1. Compute the largest $\bar{\alpha}$ such that $y + \bar{\alpha} d \in X \cap \mathcal{Z}$. Set $\alpha = \min\{\bar{\alpha}, \tilde{\alpha}\}$.
Step 2. If $\alpha > 0$ and $f(y + \alpha d) \leq f(y) - \xi$ **then** go to Step 6.
Step 3. Compute the largest $\bar{\alpha}$ such that $y - \bar{\alpha} d \in X \cap \mathcal{Z}$. Set $\alpha = \min\{\bar{\alpha}, \tilde{\alpha}\}$.
Step 4. If $\alpha > 0$ and $f(y - \alpha d) \leq f(y) - \xi$ **then** set $d \leftarrow -d$ and go to Step 6.
Step 5. Set $\alpha = 0$ and **return**.
Step 6. While $(\alpha < \bar{\alpha} \text{ and } f(y + 2\alpha d) \leq f(y) - \xi)$
$\alpha \leftarrow 2\alpha$.
Step 7. Set $\alpha \leftarrow \min\{\bar{\alpha}, \alpha\}$ and **return**.

and δ which are used, respectively, in the sufficient reduction criterion and for the expansion of the step.

At every iteration k Algorithm DFL, starting from the current iterate x_k, explores all the coordinate directions and produces the intermediate points y_k^i, $i = 1, \ldots, n$. When $i \in I_c$, that is for the continuous variables, the actual steps α_k^i are computed and the tentative steps $\tilde{\alpha}_k^i$ are updated as described in [15]. When $i \in I_z$, the algorithm performs a Discrete search which is very similar to the Continuous search procedure except for the fact that the sufficient reduction is governed by the parameter ξ_k. The updating formula of tentative steps $\tilde{\alpha}_k^i$ is such that $1 \leq \tilde{\alpha}_k^i \in \mathbb{Z}$.

Lemma 7 *Let* $\{x_k\}$, $\{\xi_k\}$, $\{y_k^i\}$, $\{\alpha_k^i\}$, $\{\tilde{\alpha}_k^i\}$, $i = 1, \ldots, n$, *the sequences produced by Algorithm DFL. Then,*

(i) *Algorithm DFL is well-defined*;
(ii) *for all* $i \in I_c$

$$\lim_{k \to \infty} \alpha_k^i = 0, \tag{8}$$

$$\lim_{k\to\infty} \tilde{\alpha}_k^i = 0; \tag{9}$$

(iii)

$$\lim_{k\to\infty} \xi_k = 0.$$

Proof In order to prove that Algorithm DFL is well defined, we have to ensure that both the Continuous search and Discrete search procedures, when performed along a direction d_k^i, with $i \in \{1, \ldots, n\}$, terminates in a finite number j of steps. This is clearly true since, by the instructions of the two procedures,

$$y_k^i + \delta^{-j}\alpha d_k^i \in X \quad \text{for all } i \in I_c,$$
$$y_k^i + 2^j\alpha d_k^i \in X \quad \text{for all } i \in I_z,$$

and X, by assumption, is a compact set.

Now we prove assertion (ii). For every $i \in I_c$, we prove (8) by splitting the iteration sequence $\{k\}$ into two parts, $K^{'}$ and $K^{''}$. We identify with $K^{'}$ those iterations where

$$\alpha_k^i = 0 \tag{10}$$

and with $K^{''}$ those iterations where $\alpha_k^i \neq 0$ is produced by the Continuous search. Then the instructions of the algorithm imply

$$f(x_{k+1}) \leq f(y_k^i + \alpha_k^i d_k^i) \leq f(y_k^i) - \gamma(\alpha_k^i)^2\|d_k^i\|^2 \leq f(x_k) - \gamma(\alpha_k^i)^2\|d_k^i\|^2. \tag{11}$$

Taking into account the compactness assumption on X, it follows from (11) that $\{f(x_k)\}$ tends to a limit $\bar{f}$. If $K^{''}$ is an infinite subset, recalling that $\|d_k^i\| = 1$ we obtain

$$\lim_{k\to\infty, k\in K''} \alpha_k^i = 0. \tag{12}$$

Therefore, (10) and (12) imply (8).

In order to prove (9), for each $i \in I_c$ we split the iteration sequence $\{k\}$ into two parts, K_1 and K_2. We identify with K_1 those iterations where the Continuous search procedure, along the direction d_k^i, returns an $\alpha_k^i > 0$, for which we have:

$$\tilde{\alpha}_{k+1}^i = \alpha_k^i. \tag{13}$$

We denote by K_2 those iterations where we have failed in decreasing the objective function along the directions d_k^i and $-d_k^i$. By the instructions of the algorithm it follows that for all $k \in K_2$

$$\tilde{\alpha}_{k+1}^i \leq \theta\tilde{\alpha}_k^i, \tag{14}$$

where $\theta \in (0, 1)$.

If K_1 is an infinite subset, from (13) and (8) we get that

$$\lim_{k\to\infty, k\in K_1} \tilde{\alpha}_{k+1}^i = 0. \tag{15}$$

Now, let us assume that K_2 is an infinite subset. For each $k \in K_2$, let m_k (we omit the dependence from i) be the biggest index such that $m_k < k$ and $m_k \in K_1$. Then we have:

$$\tilde{\alpha}^i_{k+1} \leq \theta^{(k+1-m_k)} \tilde{\alpha}^i_{m_k} \leq \tilde{\alpha}^i_{m_k} \tag{16}$$

(we can assume $m_k = 0$ if the index m_k does not exist, that is, K_1 is empty).

As $k \to \infty$ and $k \in K_2$, either $m_k \to \infty$ (namely, K_1 is an infinite subset) or $(k + 1 - m_k) \to \infty$ (namely, K_1 is finite). Hence, if K_2 is an infinite subset, (16) together with (15), or the fact that $\theta \in (0, 1)$, yields

$$\lim_{k \to \infty, k \in K_2} \tilde{\alpha}^i_{k+1} = 0, \tag{17}$$

so that (9) is proved, and this concludes the proof of point (ii).

Now we prove point (iii). By the instruction of Algorithm DFL the sequence $\{\xi_k\}$ is monotonically non-increasing, that is, $0 < \xi_{k+1} \leq \xi_k$, for all k. Hence $\{\xi_k\}$ converges to a limit $M \geq 0$. Let us suppose, by contradiction, that $M > 0$. If this were the case, then an index $\bar{k} > 0$ would exist such that $\xi_{k+1} = \xi_k = M$ for all $k \geq \bar{k}$. Moreover, for every index $k \geq \bar{k}$, a index $\bar{\imath} \in I_z$ (possibly depending on k) would exist such that

$$f(x_{k+1}) \leq f(y^{\bar{\imath}}_k \pm \alpha^{\bar{\imath}}_k d^{\bar{\imath}}_k) \leq f(y^{\bar{\imath}}_k) - M \leq f(x_k) - M, \tag{18}$$

otherwise the algorithm would have set $\xi_{k+1} = \theta \xi_k$. Relation (18) implies $f(x_k) \to -\infty$ thus contradicting the assumption that f is continuous on the compact set X, and this concludes the proof. □

By Point (iii) of the preceding proposition and the updating rule of parameter ξ_k in Algorithm DFL, it follows that the set

$$H = \{k : \xi_{k+1} < \xi_k\}$$

is infinite.

Proposition 8 *Let $\{x_k\}$ be the sequence of points produced by Algorithm DFL. Let $H \subseteq \{1, 2, \ldots\}$ be defined as in Lemma 7 and x^* be any accumulation point of $\{x_k\}_H$, then*

$$\nabla_c f(x^*)^T (x - x^*)_c \geq 0, \quad \textit{for all } x \in X.$$

Proof For any accumulation point x^* of $\{x_k\}_H$, let $K \subseteq H$ be an index set such that

$$\lim_{k \to \infty, k \in K} x_k = x^*. \tag{19}$$

Let us note that, by the instructions of Algorithm DFL, for all $k \in K$, $(y^{n+1}_k)_z = (x_k)_z$ and $\tilde{\alpha}^i_k = 1$, $i \in I_z$. Hence, for all $k \in K$, by recalling (19), the discrete variables are no longer updated. Then the proof follows by analogous reasoning as in [15]. □

Proposition 9 *Let* $\{x_k\}$, $\{\xi_k\}$, $\{y_k^i\}$, $i = 1, \ldots, n$, *be the sequence of points produced by Algorithm DFL. Let* $H \subseteq \{1, 2, \ldots\}$ *be defined as in Lemma 7 and* x^* *be any accumulation point of* $\{x_k\}_H$, *then*

$$f(x^*) \leq f(\bar{x}), \quad \textit{for all } \bar{x} \in \mathcal{N}_z(x^*) \cap X.$$

Proof Let $K \subseteq H$ be an index set such that

$$\lim_{k \to \infty, k \in K} x_k = x^*.$$

For every $k \in H$, it results

$$(y_k^{n+1})_z = (x_k)_z,$$
$$\tilde{\alpha}_k^i = 1, \quad i \in I_z,$$

that is, no discrete variable is updated by the Discrete search procedure.

Let us consider any point $\bar{x} \in \mathcal{N}_z(x^*) \cap X$. Then a direction $\bar{d} \in D(x^*) \cap D^z$ exists such that

$$\bar{x} = x^* + \bar{d}. \tag{20}$$

Recalling the definition of y_k^i in Algorithm DFL and the definition of the discrete neighborhood $\mathcal{N}_z(x)$, we have, for all $k \in H$ and sufficiently large, that

$$(x^*)_z = (x_k)_z = (y_k^i)_z, \quad i = 1, \ldots, n.$$

Further, by Lemma 7, we have

$$\lim_{k \to \infty, k \in K} y_k^i = x^*, \quad i = 1, \ldots, n.$$

Then, (20) implies

$$(x_k + \bar{d})_j = (y_k^i + \bar{d})_j = (x^* + \bar{d})_j = (\bar{x})_j, \quad i = 1, \ldots, n, j \in I_z.$$

Now, Proposition 6 guarantees that for $k \in K$ and sufficiently large, a direction $d_k^{\bar{\imath}} \in D(x_k) \cap D^z$ exists such that $d_k^{\bar{\imath}} = \bar{d}$, so that

$$(x_k + d_k^{\bar{\imath}})_j = (y_k^{\bar{\imath}} + d_k^{\bar{\imath}})_j = (x^* + d_k^{\bar{\imath}})_j = (\bar{x})_j, \quad j \in I_z,$$

for all $k \in K$ and sufficiently large. Hence, for k sufficiently large and $k \in K$,

$$y_k^{\bar{\imath}} + d_k \bar{\imath} \in X \cap \mathbb{Z}.$$

Then, we have

$$f(y_k^{\bar{\imath}} + d_k^{\bar{\imath}}) > f(y_k^{\bar{\imath}}) - \xi_k. \tag{21}$$

Now, by (iii) of Lemma 7, and taking the limit for $k \to \infty, k \in K$ in (21), the result follows. □

Theorem 10 *Let $\{x_k\}$ be the sequence of points generated by Algorithm DFL. Let $H \subseteq \{1, 2, \ldots\}$ be defined as in Lemma 7. Then,*

(i) *a limit point of $\{x_k\}_H$ exists;*
(ii) *every limit point x^* of $\{x_k\}_H$ is stationary for Problem (1).*

Proof Point (i). Since $\{x_k\}_H$ belongs to the compact set X, it admits limit points. The prove of point (ii) follows by considering Propositions 8 and 9. □

3.2 A partially distributed algorithmic scheme

In this subsection we introduce a partially distributed Algorithm DFL$_{\text{ord}}$ for bound constrained nonlinear mixed variable programming problems. The algorithm consists of two distinct phases. In the first one the continuous variables are updated by means of distributed linesearches thus returning a point $\tilde{y}$. Then, in the second phase, starting from $\tilde{y}$, the directions related to the discrete variables are investigated and the point that yields the best objective function reduction is returned. Differently from Algorithm DFL, the discrete search procedure only requires a simple reduction of the objective function. Algorithm DFL$_{\text{ord}}$ has slightly stronger convergence properties than DFL. Indeed, it is possible to show that every limit point of the sequence of iterates generated by the algorithm is stationary for Problem (1). However, these stronger convergence properties are balanced by a reduced flexibility in the discrete variables exploration.

Proposition 11 *Let $\{x_k\}$ be the sequence of points produced by Algorithm DFL$_{\text{ord}}$ and x^* be an accumulation point. Then*

$$\nabla_c f(x^*)^T (x - x^*)_c \geq 0, \quad \text{for all } x \in X.$$

Proof The proof follows by analogous reasoning as in reference [15]. □

Proposition 12 *Let $\{x_k\}$ be the sequence of points produced by Algorithm DFL$_{\text{ord}}$ and x^* be an accumulation point. Then,*

$$f(x^*) \leq f(\bar{x}),$$

for all $\bar{x} \in \mathcal{N}(x^*) \cap X$.

Proof Let K be an index set such that

$$\lim_{k \to \infty, k \in K} x_k = x^*. \tag{22}$$

By the instructions of Algorithm DFL$_{\text{ord}}$, we know that the sequence $\{f(x_k)\}$ is monotonically non-increasing. Since by the stated assumptions the objective function is bounded from below, we have that $\{f(x_k)\}$ is convergent to a limit f^*. Then, we also have that

$$\lim_{k \to \infty, k \in K} f(x_k) = f(x^*) = f^*.$$

Algorithm DFL$_{\text{ord}}$

Data. $\theta \in (0,1)$, $x_0 \in X \cap \mathcal{Z}$, $\tilde{\alpha}_0^i > 0$, $i \in I_c$, $\tilde{\alpha}_0^i = 1$, $i \in I_z$, and set $d_0^i = e^i$, for $i = 1, \ldots, n$.

$I_c = \{1, \ldots, r\}$, $I_z = \{r+1, \ldots, n\}$.

For $k = 0, 1, \ldots$

Set $y_k^1 = x_k$.

For $i = 1, \ldots, r$

compute α by the *Continuous search* $(\tilde{\alpha}_k^i, y_k^i, d_k^i; \alpha)$

If $\alpha = 0$ **then** set $\alpha_k^i = 0$ and $\tilde{\alpha}_{k+1}^i = \theta \tilde{\alpha}_k^i$.
else set $\alpha_k^i = \alpha$, $\tilde{\alpha}_{k+1}^i = \alpha$.

Set $y_k^{i+1} = y_k^i + \alpha_k^i d_k^i$ and $d_{k+1}^i = d_k^i$.

End For

For $i = r+1, \ldots, n$

compute α by the *Discrete Search*$(\tilde{\alpha}_k^i, y_k^{r+1}, d_k^i, 0; \alpha)$

If $\alpha = 0$ **then** set $\alpha_k^i = 0$ and $\tilde{\alpha}_{k+1}^i = \max\{1, \lfloor \tilde{\alpha}_k^i / 2 \rfloor\}$.
else set $\alpha_k^i = \alpha$, $\tilde{\alpha}_{k+1}^i = \alpha$.

Set $y_k^{i+1} = y_k^{r+1} + \alpha_k^i d_k^i$ and $d_{k+1}^i = d_k^i$.

End For

Set $\tilde{x}_{k+1} = \underset{y \in \{y_k^{r+1}, \ldots, y_k^{n+1}\}}{\arg\min} f(y)$.

Find $x_{k+1} \in X \cap \mathcal{Z}$ such that $f(x_{k+1}) \leq f(\tilde{x}_{k+1})$.

End For

Let us consider any point $\bar{x} \in N(x^*) \cap X$. Then a direction $\bar{d} \in D(x^*) \cap D^z$ exists such that

$$\bar{x} = x^* + \bar{d}.$$

We suppose, by contradiction that $\bar{d}$ is such that

$$f(x^* + \bar{d}) = f(x^*) - \delta < f(x^*), \tag{23}$$

with $\delta > 0$. Let $\epsilon > 0$, and consider the neighborhood $\mathcal{B}(x^*; \epsilon)$ such that

$$|f(x) - f(x^*)| \leq \delta/2,$$
$$|f(x + \bar{d}) - f(x^* + \bar{d})| \leq \delta/2,$$

for all $x \in \mathcal{B}(x^*; \epsilon) \cap X$.

By considering (22) and point (ii) of Lemma 7, we have that, for $k \in K$ and sufficiently large, $y_k^{r+1} \in \mathcal{B}(x^*; \epsilon) \cap X$. Further, Proposition 6 guarantees that for $k \in K$ and sufficiently large, an index $\bar{j} > r$ exists such that a direction $d_k^{\bar{j}} \in D(x_k) \cap D^z$ can be found such that $d_k^{\bar{j}} = \bar{d}$ and $y_k^{\bar{j}+1}$ satisfies

$$|f(y_k^{\bar{j}+1}) - f(x^* + \bar{d})| \leq \delta/2. \tag{24}$$

The above relation along with (23) implies that

$$f(y_k^{\bar{j}+1}) \leq f(x^*) - \delta/2.$$

Then, Algorithm DFL$_{\text{ord}}$ would generate the new iterate x_{k+1} such that $f(x_{k+1}) \leq f(x^*) - \delta/2$, thus contradicting the fact that $\{f(x_k)\}$ is convergent toward $f(x^*) = f^*$. □

Theorem 13 *Let $\{x_k\}$ be the sequence of points generated by Algorithm DFL$_{\text{ord}}$. Then,*

(i) *a limit point of $\{x_k\}$ exists;*
(ii) *every limit point of $\{x_k\}$ is stationary for Problem (1).*

Proof The proof of point (i) follows by considering that $\{x_k\}$ belongs to the set X which is compact by assumption. Point (ii) follows by considering Propositions 11 and 12. □

3.3 An algorithm converging toward strong stationary points

In this subsection we propose another algorithm for the solution of Problem (1) and we prove that it is convergent to strong stationary points. In order to guarantee this stronger convergence property, a deeper investigation of the discrete neighborhood is carried out by a so-called "local search" procedure. The local search procedure first performs a linesearch along the direction related to a discrete variable. Then, if a point yielding a sufficient decrease of the objective function is found, it becomes the current point. Otherwise, if a point z is found which is promising, that is, not significantly worse in function value than the current point, a distributed search is performed starting from z.

Algorithm SDFL along with the Local search procedure, generates some sequences and, in particular, the following sequences: $\{x_k\}$, $\{\xi_k\}$, $\{y_k^i\}$, $\{d_k^i\}$, $\{\alpha_k^i\}$, $\{\tilde{\alpha}_k^i\}$, $\{z_k^i\}$, for $i = 1, \ldots, n$. Moreover, we remark that the Local search procedure can be viewed as a Discrete search enriched by a Grid search. More precisely, the Grid search is used to better explore the neighborhood of a promising point z with respect to the current point y, that is a point z such that $f(y) - \xi \leq f(z) < f(y) + \nu$.

Lemma 14 *The Local search procedure is well-defined.*

Proof In order to prove that procedure Local search is well-defined, we need to show that the condition at Step 3 is eventually satisfied. Let us assume, by contradiction,

Algorithm SDFL

Data. $\theta \in (0,1)$, $\xi_0 > 0$, $x_0 \in X \cap \mathcal{Z}$, $\tilde{\alpha}_0^i > 0$, $i \in I_c$, $\tilde{\alpha}_0^i = 1$, $i \in I_z$, and set $d_0^i = e^i$, for $i = 1, \ldots, n$.

For $k = 0, 1, \ldots$

Set $y_k^1 = x_k$.

For $i = 1, \ldots, n$

If $i \in I_c$ **then** compute α by the *Continuous search*$(\tilde{\alpha}_k^i, y_k^i, d_k^i; \alpha)$

If $\alpha = 0$ **then** set $\alpha_k^i = 0$ and $\tilde{\alpha}_{k+1}^i = \theta \tilde{\alpha}_k^i$.

else set $\alpha_k^i = \alpha$, $\tilde{\alpha}_{k+1}^i = \alpha$.

Set $d_{k+1}^i = d_k^i$.

else compute α by the *Local Search*$(\tilde{\alpha}_k^i, y_k^i, d_k^i, \xi_k; \alpha, \tilde{z})$

If $\alpha = 0$ and $\tilde{z} \neq y_k^i$ **then** $\alpha_k^i = 0$, $\tilde{\alpha}_{k+1}^i = \tilde{\alpha}_k^i$, set $y_k^{n+1} = \tilde{z}$, $d_{k+1}^i = d_k^i$ and **Exit For**

If $\alpha = 0$ **then**

compute α by the *Local Search*$(\tilde{\alpha}_k^i, y_k^i, -d_k^i, \xi_k; \alpha, \tilde{z})$

If $\alpha = 0$ and $\tilde{z} \neq y_k^i$ **then** set $\alpha_k^i = 0$, $\tilde{\alpha}_{k+1}^i = \tilde{\alpha}_k^i$, $y_k^{n+1} = \tilde{z}$, $d_{k+1}^i = -d_k^i$ and **Exit For**

If $\alpha = 0$ **then** set $\alpha_k^i = 0$, $\tilde{\alpha}_{k+1}^i = \max\{1, \lfloor \tilde{\alpha}_k^i/2 \rfloor\}$ and $d_{k+1}^i = d_k^i$.

else set $\alpha_k^i = \alpha$, $\tilde{\alpha}_{k+1}^i = \alpha$ and $d_{k+1}^i = -d_k^i$.

else set $\alpha_k^i = \alpha$, $\tilde{\alpha}_{k+1}^i = \alpha$ and $d_{k+1}^i = d_k^i$.

Endif

Set $y_k^{i+1} = y_k^i + \alpha_k^i d_k^i$.

End For

If $(y_k^{n+1})_z = (x_k)_z$ and $\tilde{\alpha}_k^i = 1$, $i \in I_z$, **then** set $\xi_{k+1} = \theta \xi_k$ **else** set $\xi_{k+1} = \xi_k$.

Find $x_{k+1} \in X \cap \mathcal{Z}$ such that $f(x_{k+1}) \leq f(y_k^{n+1})$.

End For

the condition at Step 3 is never satisfied. If this was the case, then we would get a contradiction with the compactness of set X. □

Lemma 15 *Let* $\{x_k\}$, $\{\xi_k\}$, $\{y_k^i\}$, $\{\alpha_k^i\}$, $\{\tilde{\alpha}_k^i\}$, $i = 1, \ldots, n$, *be the sequences produced by Algorithm SDFL. Then,*

(i) *Algorithm SDFL is well-defined*;
(ii) *for all* $i \in I_c$

$$\lim_{k \to \infty} \alpha_k^i = 0, \tag{25}$$

Local search$(\tilde{\alpha}, y, d, \xi; \alpha, \tilde{z})$

Data. $\nu > 0$.
Initialization. Compute the largest $\bar{\alpha}$ such that $y + \bar{\alpha} d \in X \cap \mathcal{Z}$. Set $\alpha = \min\{\bar{\alpha}, \tilde{\alpha}\}$ and $z = y + \alpha d$.
Step 0. If $\alpha = 0$ or $f(z) > f(y) + \nu$ **then** Set $\tilde{z} = y$, $\alpha = 0$ and **return**.
Step 1. If $\alpha > 0$ and $f(z) \le f(y) - \xi$ **then** go to Step 2.
Else go to Step 4.
Step 2. While ($\alpha < \bar{\alpha}$ and $f(y + 2\alpha d) \le f(y) - \xi$)
$\alpha \leftarrow 2\alpha$.
Step 3. Set $\alpha \leftarrow \min\{\bar{\alpha}, \alpha\}$ and $\tilde{z} = y + \alpha d$ and **return**.
Step 4. (*Grid search*) Set $z = y + \alpha d$.
Set $w^1 = z$.
For $i = 1, \ldots, n$
Let $q^i = e^i$.
If $i \in I_z$ compute α by the *Discrete Search*$(\tilde{\alpha}^i, w^i, q^i, \xi; \alpha)$
If $\alpha \neq 0$ **and** $f(w^i + \alpha q^i) \le f(y) - \xi$ **then** set $\tilde{z} = w^i + \alpha q^i$, $\alpha = 0$ and **return**
If $i \in I_c$ compute α by the *Continuous search*$(\tilde{\alpha}^i, w^i, q^i; \alpha)$
If $\alpha \neq 0$ **and** $f(w^i + \alpha q^i) \le f(y) - \xi$ **then** set $\tilde{z} = w^i + \alpha q^i$, $\alpha = 0$ and **return**
Set $w^{i+1} = w^i + \alpha q^i$.

End For
Set $\tilde{z} = y$, $\alpha = 0$ and **return**.

$$\lim_{k \to \infty} \tilde{\alpha}_k^i = 0; \tag{26}$$

(iii)

$$\lim_{k \to \infty} \xi_k = 0.$$

Proof In order to prove that Algorithm SDFL is well defined, we have to ensure that, when performed along a direction d_k^i, with $i \in \{1, \ldots, n\}$, Step 1 and 2 of the Local search procedure are executed a finite number j of times, since by Lemma 14 we already have that the Local search procedure is well-defined. By the instructions of the Continuous search procedure, when Step 2 is executed, we have

$$y_k^i + \delta^{-j} \alpha d_k^i \in X \quad \text{for all } i \in I_c,$$

then, the proof of point (i) follows by recalling that X, by assumption, is a compact set.

Now we prove assertion (ii). For every $i \in I_c$, we prove (25) by splitting the iteration sequence $\{k\}$ into two parts, K' and K''. We identify with K' those iterations

where

$$\alpha_k^i = 0 \tag{27}$$

and with K'' those iterations where $\alpha_k^i \neq 0$ is produced by the Continuous search. Then the instructions of the algorithm imply

$$f(x_{k+1}) \leq f(y_k^i + \alpha_k^i d_k^i) \leq f(y_k^i) - \gamma(\alpha_k^i)^2 \|d_k^i\|^2 \leq f(x_k) - \gamma(\alpha_k^i)^2 \|d_k^i\|^2. \tag{28}$$

Taking into account the compactness assumption on X, it follows from (28) that $\{f(x_k)\}$ tends to a limit $\bar{f}$. If K'' is an infinite subset, recalling that $\|d_k^i\| = 1$ we obtain

$$\lim_{k\to\infty, k\in K''} \alpha_k^i = 0. \tag{29}$$

Therefore, (27) and (29) imply (25).

In order to prove (26), for each $i \in I_c$ we split the iteration sequence $\{k\}$ into two parts, K_1 and K_2. We identify with K_1 those iterations where the Continuous search procedure, along the direction d_k^i, returns an $\alpha_k^i > 0$, for which we have:

$$\tilde{\alpha}_{k+1}^i = \alpha_k^i. \tag{30}$$

We denote by K_2 those iterations where we have failed in decreasing the objective function along the directions d_k^i and $-d_k^i$. By the instructions of the algorithm it follows that for all $k \in K_2$

$$\tilde{\alpha}_{k+1}^i \leq \theta \tilde{\alpha}_k^i, \tag{31}$$

where $\theta \in (0, 1)$.

If K_1 is an infinite subset, from (30) and (25) we get that

$$\lim_{k\to\infty, k\in K_1} \tilde{\alpha}_{k+1}^i = 0. \tag{32}$$

Now, let us assume that K_2 is an infinite subset. For each $k \in K_2$, let m_k (we omit the dependence from (i) be the biggest index such that $m_k < k$ and $m_k \in K_1$. Then we have:

$$\tilde{\alpha}_{k+1}^i \leq \theta^{(k+1-m_k)} \tilde{\alpha}_{m_k}^i \leq \tilde{\alpha}_{m_k}^i \tag{33}$$

(we can assume $m_k = 0$ if the index m_k does not exist, that is, K_1 is empty).

As $k \to \infty$ and $k \in K_2$, either $m_k \to \infty$ (namely, K_1 is an infinite subset) or $(k + 1 - m_k) \to \infty$ (namely, K_1 is finite). Hence, if K_2 is an infinite subset, (33) together with (32), or the fact that $\theta \in (0, 1)$, yields

$$\lim_{k\to\infty, k\in K_2} \tilde{\alpha}_{k+1}^i = 0, \tag{34}$$

so that (26) is proved, and this concludes the proof of point (ii).

Now we prove point (iii). By the instruction of Algorithm SDFL the sequence $\{\xi_k\}$ is monotonically non-increasing, that is, $0 < \xi_{k+1} \leq \xi_k$, for all k. Hence $\{\xi_k\}$ converges to a limit $M \geq 0$. Let us suppose, by contradiction, that $M > 0$. If this was

the case, then an index $\bar{k} > 0$ would exist such that $\xi_{k+1} = \xi_k = M$ for all $k \geq \bar{k}$. Moreover, it would result

$$f(x_{k+1}) \leq f(y_k^{n+1}) \leq f(x_k) - M, \quad \text{and} \quad (y_k^{n+1})_z \neq (x_k)_z, \tag{35}$$

otherwise the algorithm would have set $\xi_{k+1} = \theta\xi_k$. Relation (35) implies $f(x_k) \to -\infty$ thus contradicting the assumption that f is continuous on the compact set X, and this concludes the proof. □

By Point (iii) of the preceding proposition and the updating rule of parameter ξ_k in Algorithm DFL, it follows that the set

$$H = \{k : \xi_{k+1} < \xi_k\}$$

is infinite.

Proposition 16 *Let* $\{x_k\}$, $\{\xi_k\}$, $\{y_k^i\}$, $\{z_k^i\}$, $\{\alpha_k^i\}$, $\{\tilde{\alpha}_k^i\}$, $i = 1, \ldots, n$, *be the sequence of points produced by Algorithm SDFL. Let* $H \subseteq \{1, 2, \ldots\}$ *be defined as in Lemma 15. Then,*

(i) *the subsequence* $\{x_k\}_H$ *admits limit points*;
(ii) *every limit point* x^* *of* $\{x_k\}_H$ *is a strong stationary point for Problem* (1).

Proof Point (i) is proved by considering that $\{x_k\}_H$ belongs to X which is compact by assumption.

Point (ii). Let x^* be a limit point of $\{x_k\}_H$ and $K \subseteq H$ be an index set such that

$$\lim_{k \to \infty, k \in K} x_k = x^*.$$

By recalling the definition of Strong Stationary Point, we have to show that x^* satisfies (3) and (4), and, for all $\bar{x} \in \mathcal{N}_z(x^*) \cap X$ such that $f(\bar{x}) = f(x^*)$, it holds that

$$\nabla_c f(\bar{x})^T (x - \bar{x})_c \geq 0, \quad \text{for all } x \in X, \tag{36}$$

$$f(\bar{x}) \leq f(x) \quad \text{for all } x \in \mathcal{N}_z(\bar{x}) \cap X. \tag{37}$$

Recalling the fact that the Local search is an enrichment of the Discrete search defined in Sect. 3.1. Hence, the limit points produced by Algorithm SDFL surely satisfy Properties (3) and (4) which can be derived by using Propositions 8 and 9.

Now we have to show that (36) and (37) hold. For any choice of $\bar{x} \in \mathcal{N}_z(x^*) \cap X$ such that $f(\bar{x}) = f(x^*)$, and, reasoning as in Proposition 9, we can find a subsequence $\{z_k^{\bar{i}}\}_K$, for some index $\bar{i} \in \{1, 2, \ldots, n\}$, such that,

$$\lim_{k \to \infty, k \in K} z_k^{\bar{i}} = \bar{x}, \tag{38}$$

and, for all $k \in K$ and sufficiently large, that

$$(\bar{x})_z = (z_k^{\bar{i}})_z.$$

Let us consider any point $\tilde{x} \in \mathcal{N}_z(\bar{x}) \cap X$. Then a direction $\tilde{d} \in D(\bar{x}) \cap D^z$ exists such that

$$\tilde{x} = \bar{x} + \tilde{d}. \tag{39}$$

Then, (39) implies

$$(z_k^{\bar{i}} + \tilde{d})_j = (\bar{x} + \tilde{d})_j = (\tilde{x})_j, \quad j \in I_z.$$

Now, Proposition 6 guarantees that for $k \in K$ and sufficiently large, a direction $d_k^{\bar{j}} \in D(z_k) \cap D^z$ exists such that $d_k^{\bar{j}} = \tilde{d}$, so that

$$(z_k^{\bar{i}} + d_k^{\bar{j}})_j = (\bar{x} + d_k^{\bar{j}})_j = (\tilde{x})_j, \quad j \in I_z,$$

for all $k \in K$ and sufficiently large. Hence, for k sufficiently large and $k \in K$,

$$z_k^{\bar{i}} + d_k^{\bar{j}} \in X \cap \mathbb{Z}.$$

Then, we have

$$f(z_k^{\bar{i}} + d_k^{\bar{j}}) > f(y_k^{\bar{i}}) - \xi_k. \tag{40}$$

Now, recalling (38) and that, by Lemma 15,

$$\lim_{k\to\infty, k\in K} y_k^{\bar{i}} = x^*, \qquad \lim_{k\to\infty, k\in K} \xi_k = 0, \tag{41}$$

relation (37) follows by taking the limit for $k \to \infty, k \in K$ in (40), and considering that, by assumption, $f(\bar{x}) = f(x^*)$.

Then we show that point $\bar{x}$ is stationary with respect to the continuous variables, that is (36) holds.

For every $k \in K$, it results that

$$f(z_k^{\bar{i}}) = f(w_k^1) \geq f(w_k^2) \geq \cdots \geq f(w_k^n) > f(y_k^{\bar{i}}) - \xi_k. \tag{42}$$

Taking the limit for $k \to \infty, k \in K$ in (42), and considering (38) and (41) and that, by assumption, $f(\bar{x}) = f(x^*)$, we obtain that

$$\lim_{k\to\infty, k\in K} f(w_k^i) = f(\bar{x}), \quad i = 1, \ldots, n. \tag{43}$$

For every $i \in I_c$ such that

$$f(w_k^i + \tilde{\alpha}_k^i q_k^i) > f(w_k^i) - \gamma(\tilde{\alpha}_k^i)^2,$$

we have that $w_k^{i+1} = w_k^i$ and, by Lemma 15, $\tilde{\alpha}_k^i \to 0$, for all $i \in I_c$.

On the other hand, for those indices $i \in I_c$ such that

$$f(w_k^i + \alpha_k^i q_k^i) \leq f(w_k^i) - \gamma(\alpha_k^i)^2,$$

we have that $w_k^{i+1} = w_k^i + \alpha_k^i q_k^i$ and $\alpha_k^i \to 0$, by (43). Hence, since $w_k^1 = z_k^{\bar{i}}$, by (38), $z_k^{\bar{i}} \to \bar{x}$ and $\tilde{\alpha}_k^i \to 0$ and $\alpha_k^i \to 0$, for $k \in K$, we have that

$$\lim_{k\to\infty, k\in K} w_k^i = \bar{x}. \tag{44}$$

Now, for k sufficiently large, $D(\bar{x}) \subseteq D(x_k)$. Since the grid search step in the Local search procedure explores, for every index i, both the directions e^i and $-e^i$. Thus, for every $i \in I_c$ and $\bar{d}^i \in D(\bar{x})$, we can define η_k as follows:

$$\eta_k = \begin{cases} \tilde{\alpha}_k^i & \text{if } f(w_k^i + \tilde{\alpha}_k^i \bar{d}^i) > f(w_k^i) - \gamma(\tilde{\alpha}_k^i)^2, \\ \frac{\alpha_k^i}{\delta} & \text{if } f\left(w_k^i + \frac{\alpha_k^i}{\delta}\bar{d}^i\right) > f(w_k^i) - \gamma\left(\frac{\alpha_k^i}{\delta}\right)^2. \end{cases} \tag{45}$$

Then we can write

$$\frac{f(w_k^i + \eta_k \bar{d}^i) - f(w_k^i)}{\eta_k} > -\gamma\eta_k.$$

By taking the limit, for $k \to \infty$ and $k \in K$, in the above relation and recalling that $\eta_k \to 0$, we obtain

$$\nabla_c f(\bar{x})^T \bar{d}^i \geq 0,$$

which, by recalling Proposition 5, concludes the proof. □

4 Numerical experience

In this section we report the results obtained using the two algorithms DFL and SDFL on a set of well-known test problems in continuous optimization which have been suitably modified by letting some variables assume only a finite number of values. In particular, for every even index i, variable $x^i \in X^i$ with

$$X^i = \left\{ l^i + h\frac{(u^i - l^i)}{10} \right\} \quad \text{for } h = 0, \ldots, 10.$$

First we use a set of test problems for local optimization [8, 10, 18]. On these problems we compare the performances of the proposed algorithms DFL, SDFL and of the state-of-the-art solver for derivative-free mixed integer nonlinear optimization NOMAD release 3.4.1 [2, 11]. We remark that NOMAD is designed for more general optimization problems, as it does not require the objective function to be continuously differentiable, and allows nonsmooth constraints. Then we use a set of test problems for global optimization [7, 9, 20] in order to highlight the differences between DFL and SDFL.

DFL and SDFL have been implemented in Fortran90. All the codes have been run on an Intel Core2 quad CPU 2.66 GHz with 4 GB main memory. DFL and SDFL have been run using a tolerance 10^{-3} in the stopping condition on the trial steps along the search directions. As for NOMAD, all the experiments have been conducted by using its default parameters except for `MIN_MESH_SIZE` $= 10^{-3}$ to be comparable with the stopping condition of DFL and SDFL.

Table 1 Comparison between Algorithms DFL, SDFL, and NOMAD

Problem	n	f_0	DFL		SDFL		NOMAD	
			f^*	nF	f^*	nF	f^*	nF
helval	3	5.068E+01	1.113E+00	66	1.113E+00	66	1.113E+00	188
biggsexp6	6	3.105E+00	5.741E-03	88	1.697E-03	935	1.884E-03	4379
gaussian	3	1.146E+00	2.595E-02	36	2.595E-02	58	2.595E-02	75
powbad	2	8.100E+09	9.998E+07	17	9.998E+07	17	9.998E+07	20
box	3	1.958E+00	4.974E-02	36	4.974E-02	46	4.974E-02	78
vardim	10	1.187E+06	9.454E+02	81	9.454E+02	81	9.454E+02	505
watson	9	1.225E+03	2.842E-01	154	2.230E-01	889	2.231E-01	3754
watson12	12	4.924E+02	1.434E-01	316	1.491E-01	1764	1.352E-01	20000
penalty1	10	3.022E+01	1.733E-01	81	1.733E-01	235	1.733E-01	1489
penalty2	4	4.968E+01	4.593E-01	33	4.593E-01	53	4.593E-01	134
penalty10	10	6.828E+02	2.582E+00	83	2.582E+00	164	2.582E+00	1118
brownbad	2	1.282E+12	2.210E+10	58	2.210E+10	62	2.210E+10	29
brownden	4	7.946E+06	1.679E+05	45	1.679E+05	45	1.679E+05	77
gulf	3	2.891E+01	5.025E-01	100	5.025E-01	100	5.025E-01	212
trig	10	4.843E+01	2.432E-03	196	2.007E-03	905	2.009E-03	3552
banex	2	9.252E+01	2.000E-02	22	2.000E-02	27	2.000E-02	94
powellex	4	4.369E+01	1.210E+00	37	1.210E+00	73	1.210E+00	181
beale	2	2.706E+01	2.240E-02	34	2.240E-02	57	2.241E-02	73
wood	4	1.902E+02	1.639E+00	39	1.639E+00	74	1.639E+00	391
hs1	2	3.034E+03	1.257E+00	45	7.589E-01	50	1.724E-01	39
hs2	2	6.340E+02	1.257E+00	42	1.257E+00	46	5.453E+00	15
hs3	2	1.000E-03	0.000E+00	39	0.000E+00	43	0.000E+00	39
hs4	2	3.199E+00	2.667E+00	10	2.667E+00	13	2.667E+00	10
hs5	2	5.125E+02	7.711E-01	48	7.711E-01	48	7.838E-01	21
hs25	3	3.283E+01	2.250E-03	454	2.250E-03	468	1.197E-02	571
hs38	4	4.200E+01	4.154E+01	80	4.154E+01	80	4.156E+01	76
hs45	5	1.969E+00	1.000E+00	50	1.000E+00	65	1.000E+00	275
hs229	2	1.000E+00	7.711E-01	33	7.711E-01	33	7.731E-01	28
hs242	3	7.660E+01	0.000E+00	45	0.000E+00	50	0.000E+00	125
hs247	3	8.100E+01	4.930E-30	55	4.930E-30	55	6.250E-02	34
hs257	4	4.200E+01	3.571E+01	90	3.571E+01	90	3.573E+01	51
hs307	2	2.567E+02	1.935E+02	29	1.935E+02	29	2.567E+02	9
hs328	2	1.770E+00	1.744E+00	28	1.744E+00	57	1.744E+00	36
hs357	4	1.000E+20	5.700E+02	388	4.000E-01	297	3.635E-01	814
hs358	5	1.227E+01	2.832E-01	183	2.727E-01	855	1.387E-01	5399
hs368	8	0.000E+00	-1.000E+00	63	-1.000E+00	230	-1.000E+00	583

In the following tables we report, besides the problem name, number of variables (n), objective function value on the starting point (f_0), and, for every code, the at-

Table 2 Comparison between Algorithms DFL, SDFL on a set of global optimization problems

Problem	n	f_0	DFL		SDFL	
			f^*	nF	f^*	nF
hartman 6	6	-5.0531499E-01	-3.3028103E+00	145	-3.3028151E+00	967
shekel 10	4	-8.6461583E-01	-3.8345346E+00	80	-2.8710205E+00	213
Ackley	5	1.7887799E+01	3.3009720E+00	139	3.3009720E+00	159
Ackley	10	1.7887799E+01	3.6222893E+00	277	3.0822555E+00	830
Ackley	20	1.7887799E+01	3.6222892E+00	591	3.0822555E+00	5961
Michalewics	5	0.0000000E+00	0.0000000E+00	9	0.0000000E+00	33
Michalewics	10	0.0000000E+00	0.0000000E+00	21	-2.0772108E+00	931
Michalewics	20	0.0000000E+00	-1.8861830E-01	166	-3.8945454E+00	4491
Rastrigin	5	1.5190125E+02	8.8440377E+01	126	8.8440377E+01	126
Rastrigin	10	3.0380249E+02	1.8130265E+02	230	1.8130265E+02	230
Rastrigin	20	6.0760498E+02	3.6260529E+02	465	3.6260529E+02	465
15n min.	5	5.0000001E-01	1.9625382E-01	128	1.9625382E-01	424
15n min.	10	1.0000000E+00	9.8882650E-02	243	9.8882650E-02	1132
15n min.	20	2.0000000E+00	9.8882650E-02	493	9.8882650E-02	3662

tained function value (f^*) and the number of function evaluations required to satisfy the stopping condition (nF).

From Table 1, we can notice that Algorithm DFL outperforms both SDFL and NOMAD in terms of function evaluations. Furthermore, in terms of final objective function values SDFL and NOMAD are comparable and perform slightly better than DFL, but SDFL is less expensive, in terms of function evaluations, than NOMAD. In order to better assess the improved convergence properties of algorithm SDFL with respect to DFL, we run the two codes on a set of global optimization problems. In Table 2, we report such a comparison. Both methods perform identically on the *Rastrigin* problems. On all other test problems SDFL requires more function evaluations than DFL. On one problem the SDFL solution is slightly worse than that of DFL, but it is significantly better on five test problems. In summary, Tables 1 and 2 reveal that SDFL often outperforms DFL at the expense of additional function calls. The expense of a higher number of function evaluations.

5 Conclusions

In this paper we have addressed the bound constrained mixed integer problem. First, we have defined stationary and strong stationary points and then we have proposed different algorithms and proved their convergence properties. The first two algorithms converge toward stationary points whereas the last algorithm converges toward strong stationary points. All of the proposed algorithms are of the linesearch-type. Along the continuous variables we adopt a well-studied linesearch with sufficient decrease strategy. The algorithms differ in the local search procedure that is used to update the discrete variables. A common feature of the three algorithms is that they explore

a discrete neighborhood of points whose structure is not defined a priori but it is adaptively determined by a linesearch-type procedure. Algorithm DFL performs a minimization distributed along all the variables, in the sense that the current point is updated as soon as a point yielding a sufficient reduction of the objective function is found. Algorithm DFL_{ord}, first approximately minimizes the objective function with respect to the continuous variables. Then, the directions related to the discrete variables are investigated and the point that yields the best objective function reduction is returned. In this sense, DFL_{ord} can be seen as a partially distributed linesearch algorithm. Finally, Algorithm SDFL, in order to guarantee stronger convergence properties, performs a deeper investigation of the discrete neighborhood by a local search procedure. In the paper, we also carried out a numerical experience with the algorithms DLF and SDFL and compared them with the well-known software package NOMAD. The results show the good behavior of the proposed algorithms and highlight the usefulness of the improved convergence properties of Algorithm SDFL.

As concerns future developments, we aim at extending the proposed approach to tackle the presence of general nonlinear constraints.

Acknowledgements Work partially funded by the UE (ENIAC Joint Undertaking) in the MODERN project (ENIAC-120003).

References

1. Abramson, M.A., Audet, C., Chrissis, J.W., Walston, J.G.: Mesh adaptive direct search algorithms for mixed variable optimization. Optim. Lett. **3**(1), 35–47 (2009)
2. Abramson, M.A., Audet, C., Couture, G., Dennis, J.E. Jr., Le Digabel, S.: The NOMAD project. Software available at http://www.gerad.ca/nomad
3. Abramson, M.A., Audet, C., Dennis, J.E. Jr.: Filter pattern search algorithms for mixed variable constrained optimization problems. Pac. J. Optim. **3**(3), 477–500 (2007)
4. Audet, C., Dennis, J.E. Jr.: Mesh adaptive direct search algorithms for constrained optimization. SIAM J. Optim. **17**(1), 188–217 (2006)
5. Audet, C., Dennis, J.E. Jr.: A pattern search filter method for nonlinear programming without derivatives. SIAM J. Optim. **14**(4), 980–1010 (2004)
6. Audet, C., Dennis, J.E. Jr.: Pattern search algorithms for mixed variable programming. SIAM J. Optim. **11**(3), 573–594 (2001)
7. Dixon, L.C.W., Szegö, G.P.: Towards Global Optimization, vol. 2. North Holland, Amsterdam (1975)
8. Elster, C., Neumaier, A.: A grid algorithm for bound-constrained optimization of noisy functions. IMA J. Numer. Anal. **15**, 585–608 (1995)
9. Floudas, C.A., Pardalos, P.M., Adjiman, C.S., Esposito, W.R., Gumus, Z., Harding, S.T., Klepeis, J.L., Meyer, C.A., Schweiger, C.A.: Handbook of Test Problems for Local and Global Optimization. Kluwer Academic, Dordrecht (1999)
10. Hock, W., Schittkowski, K.: Test Examples for Nonlinear Programming Codes. Lecture Notes in Economics and Mathematical Systems, vol. 187. Springer, Berlin (1981)
11. Le Digabel, S.: Algorithm 909: NOMAD: Nonlinear optimization with the MADS algorithm. ACM Trans. Math. Softw. **37**(4) (2010)
12. Lin, C.-J., Lucidi, S., Palagi, L., Risi, A., Sciandrone, M.: A decomposition algorithm model for singly linearly constrained problems subject to lower and upper bounds. J. Optim. Theory Appl. **141**, 107–126 (2009)
13. Liuzzi, G., Lucidi, S., Sciandrone, M.: Sequential penalty derivative-free methods for nonlinear constrained optimization. TR 01/2009 Dip. di Sistemi e Informatica, Università degli Studi di Firenze (2009)
14. Lucidi, S., Piccialli, V., Sciandrone, M.: An algorithm model for mixed variable programming. SIAM J. Optim. **14**, 1057–1084 (2005)

15. Lucidi, S., Sciandrone, M.: A derivative-free algorithm for bound constrained optimization. Comput. Optim. Appl. **21**(2), 119–142 (2002)
16. Lucidi, S., Sciandrone, M., Tseng, P.: Objective-derivative-free methods for constrained optimization. Math. Program. **92**(1), 37–59 (2002)
17. Lewis, R.M., Torczon, V.: Pattern search algorithms for bound constrained minimization. SIAM J. Optim. **9**(4), 1082–1099 (1999)
18. Schittkowski, K.: More Test Examples for Nonlinear Programming Codes. Lecture Notes in Economics and Mathematical Systems, vol. 282. Springer, Berlin (1987)
19. Torczon, V.: On the convergence of pattern search algorithms. SIAM J. Optim. **7**(1), 1–25 (1997)
20. Törn, A., Zilinskas, A.: Global Optimization. Springer, Berlin (1989)